Marble Labyrinth

Iordache Tiberiu-Mihai 336CA
tiberiu.iordache00@stud.acs.upb.ro

Introducere

Ce face?

Proiectul reprezintă un joc în care trebuie să navighezi o bilă prin labirint pentru a ajunge la punctul destinație într-un timp limită. Planul jocului este rotit prin servo motoare operate de un joystick, iar atingerea obiectivului va fi detectată de un senzor ultrasonic.

Scopul proiectului

Scopul proiectului este atât recreativ, pentru a aduce divertisment în timpul liber, cât și educativ din punct de vedere al explorării capabilităților plăcii Arduino Uno.

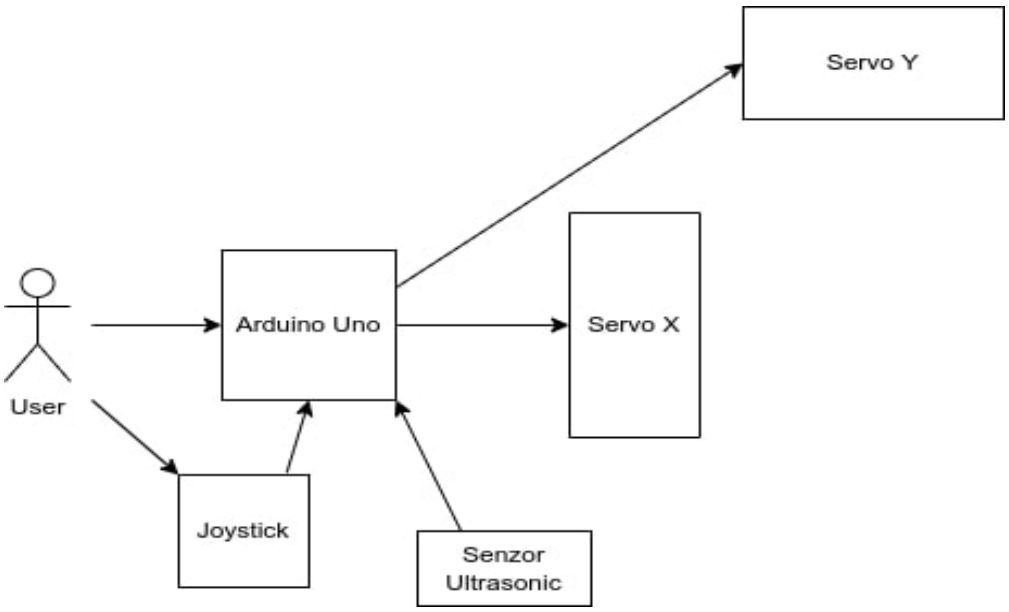
Ideea de la care am pornit

Mi-am dorit să realizez un proiect care să îmi stimlueze imaginația în construcția sa, atât din punct de vedere hardware, cât și software.

Utilitate

Proiectul poate fi folosit de oricine dorește să exploreze funcționalitățile plăcii Arduino, dezvoltarea proiectelor utilizând servo motoare sau implementarea low-level a unui timer, într-un mod distractiv și constructiv.

Descriere generală



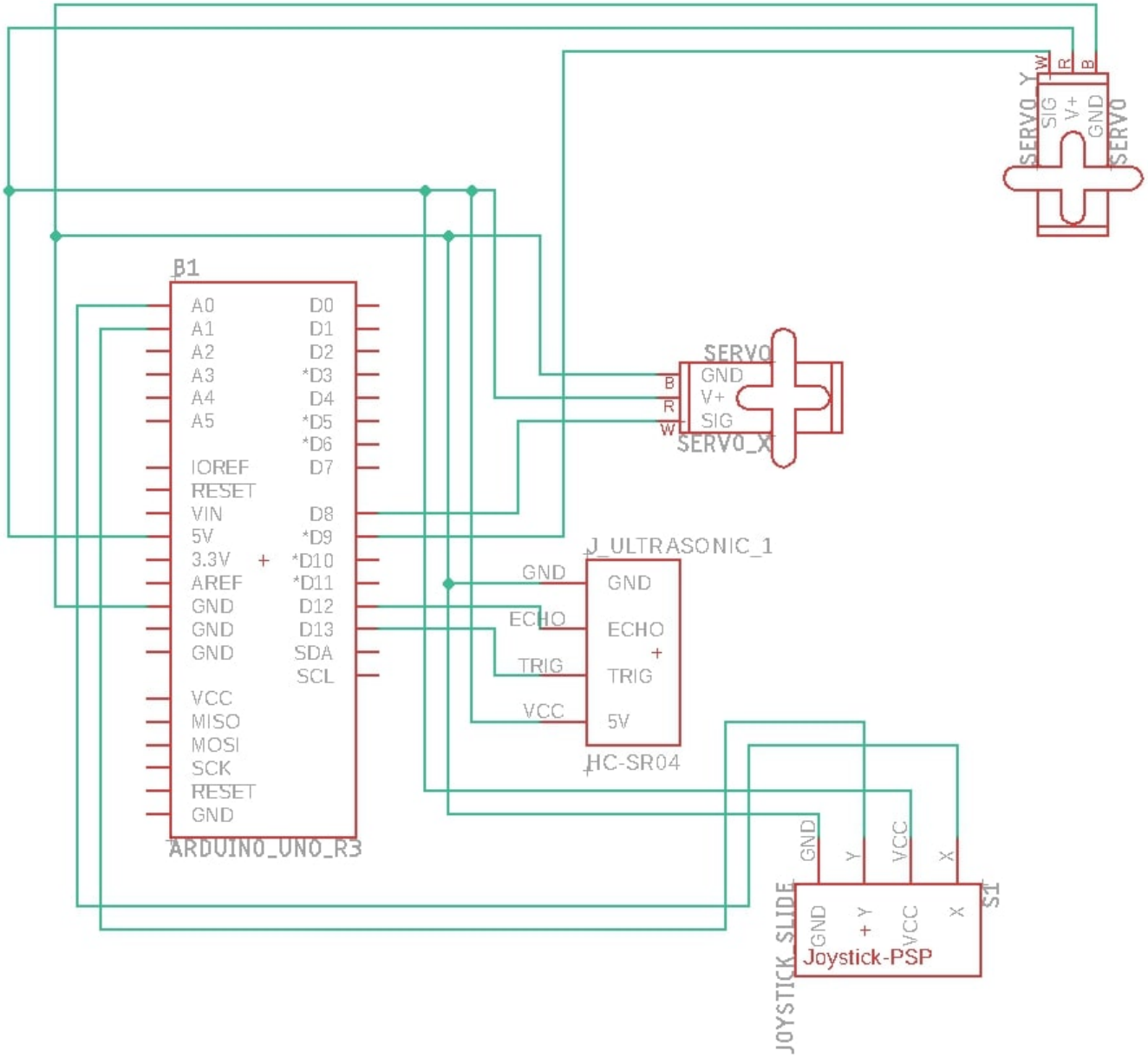
Jucătorul va interacționa cu ansamblul prezentat prin interfața serială a plăcii Arduino care primește comenzi pentru a selecta diverse moduri de joc sau pentru a reseta jocul. Odată ce începe jocul, două servo motoare vor controla axele X și Y ale planului prin intermediul unui joystick. Dacă modul de joc selectat este contra timp se va folosi și un senzor ultrasonic care va detecta când bila ajunge la destinație.

Hardware Design

Listă de piese:

- Arduino Uno R3 ATmega328P
- Breadboard
- Joystick
- 2 x Motor SERVO SG90 9G
- Senzor Ultrasonic HC-SR04
- Fire de legătură (tată-tată și mamă-tată)

Schema electrică



Software Design

Ca mediu de dezvoltare am folosit Arduino IDE 1.8.19 pentru scrierea de cod, pentru schema bloc am folosit draw.io, iar pentru schema electrică am folosit Eagle.

În cadrul implementarii am folosit biblioteca **Servo.h** pentru a controla motoarele.

În **setup()**:
Inițializez motoarele și le setez la pozitia de start(aceasta poate să fie modificată din cod prin intermediul unor variabile pentru a acomoda erorile fizice ale ansamblului). Totodata, aici realizez și configurarea timer-ului 2 și afisez la Serial Monitor mesajul de întâmpinare.

Am ales să folosesc timer2 deoarece timer0 ar fi interacționat cu functiile delay(), millis(), etc., iar timer1 ar fi interacționat cu functiile de servo. Timer2 este configurat astfel încât să genereze o întrerupere la fiecare 0,016ms. În cadrul fiecărei întreruperi voi contoriza o variabilă volatilă pentru incrementări de 0.5 asupra variabilei care ține evidența timpului rămas.

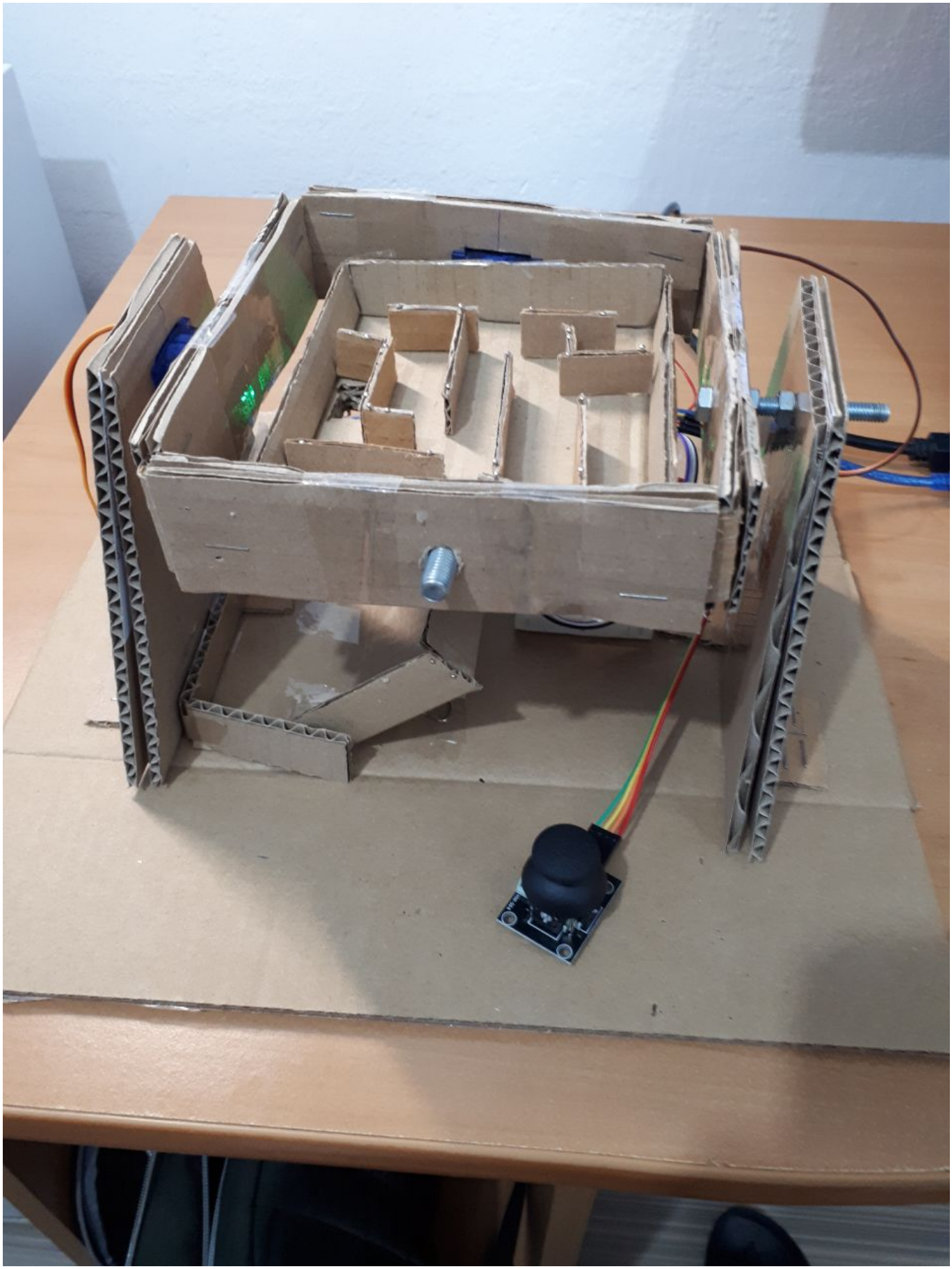
În **loop()**:
Se pot citii comenzi de la Serial atunci când niciun joc nu este activ. Atunci cand un mod de joc este pornit se vor citii date de la joystick pentru incrementa sau decrementa unghiurile celor doua servo motoare.
Deoarece tot în cadrul acestei funcții se verifică și valoarea variabilei ce ține evidența timpului rămas, nu am putut introduce apeluri de delay() după fiecare apel de servo.write(). Așadar, am decis să folosesc metoda **Debouncing** contorizând timpul trecut de la ultimul apel de servo.write() și verificând ca acesta să fie mai mare decât un delay cu valoarea de 150ms(în datasheet era menționată viteza de operare ca fiind 0.12 sec/60 degrees).
Pentru verificarea detectării unui obiect de către senzorul ultrasonic acesta a fost programat să trimită semnale de Trig și să recepționeze semnalele de Echo calculand distanța. Dacă distanța obținută se afla într-un interval setat (min, max](care poate fi modificat în cod în funcție de erorile fizice ale ansamblului) se va declanșa semnalul de WIN, iar la Serial Monitor va fi afișat un mesaj sugestiv.
În cazul în care se depășește timpul limită setat la Serial Monitor va fi afișat un mesaj sugestiv alături de lista de comenzi posibile.

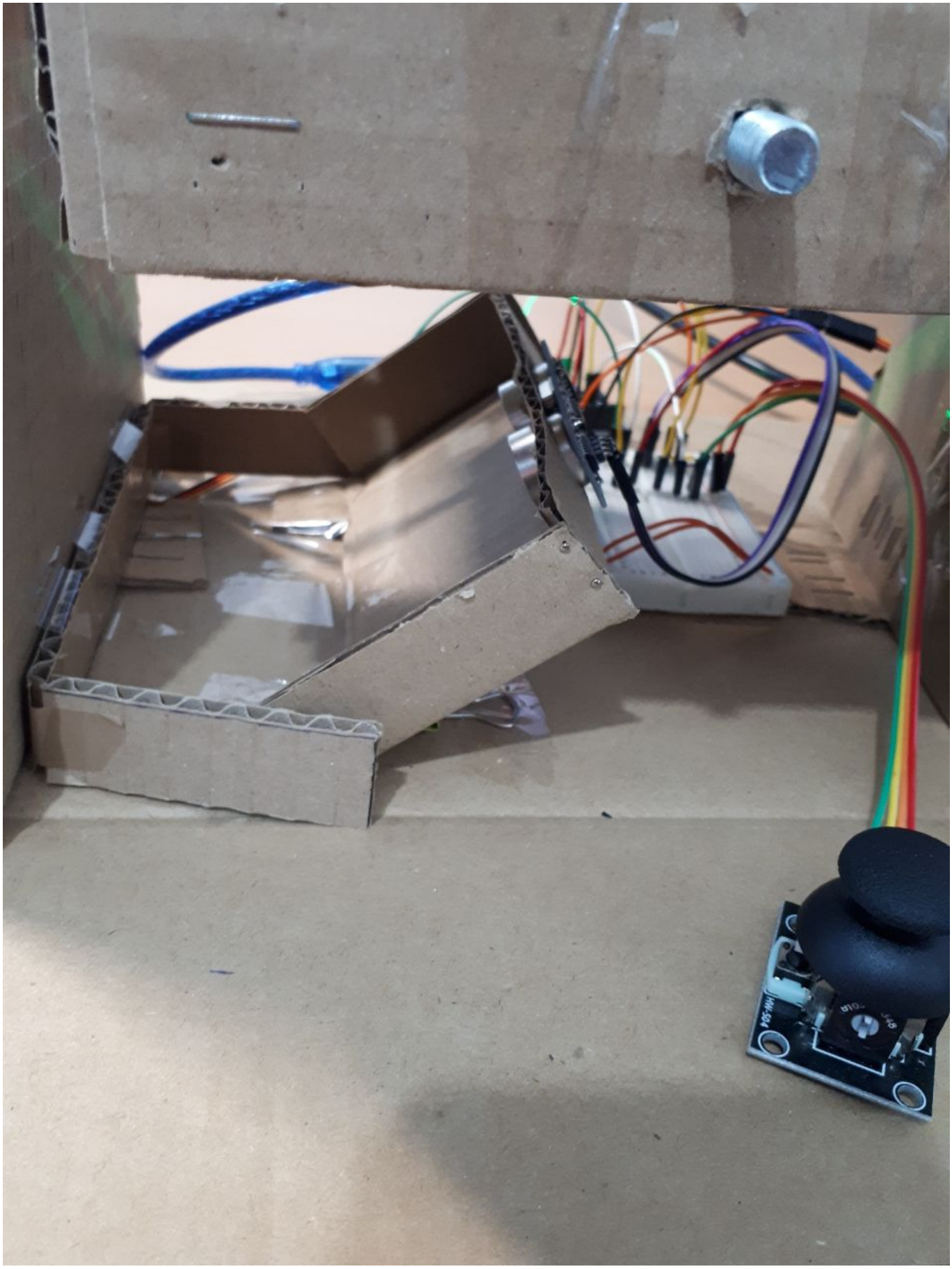
Aceste comenzi sunt:

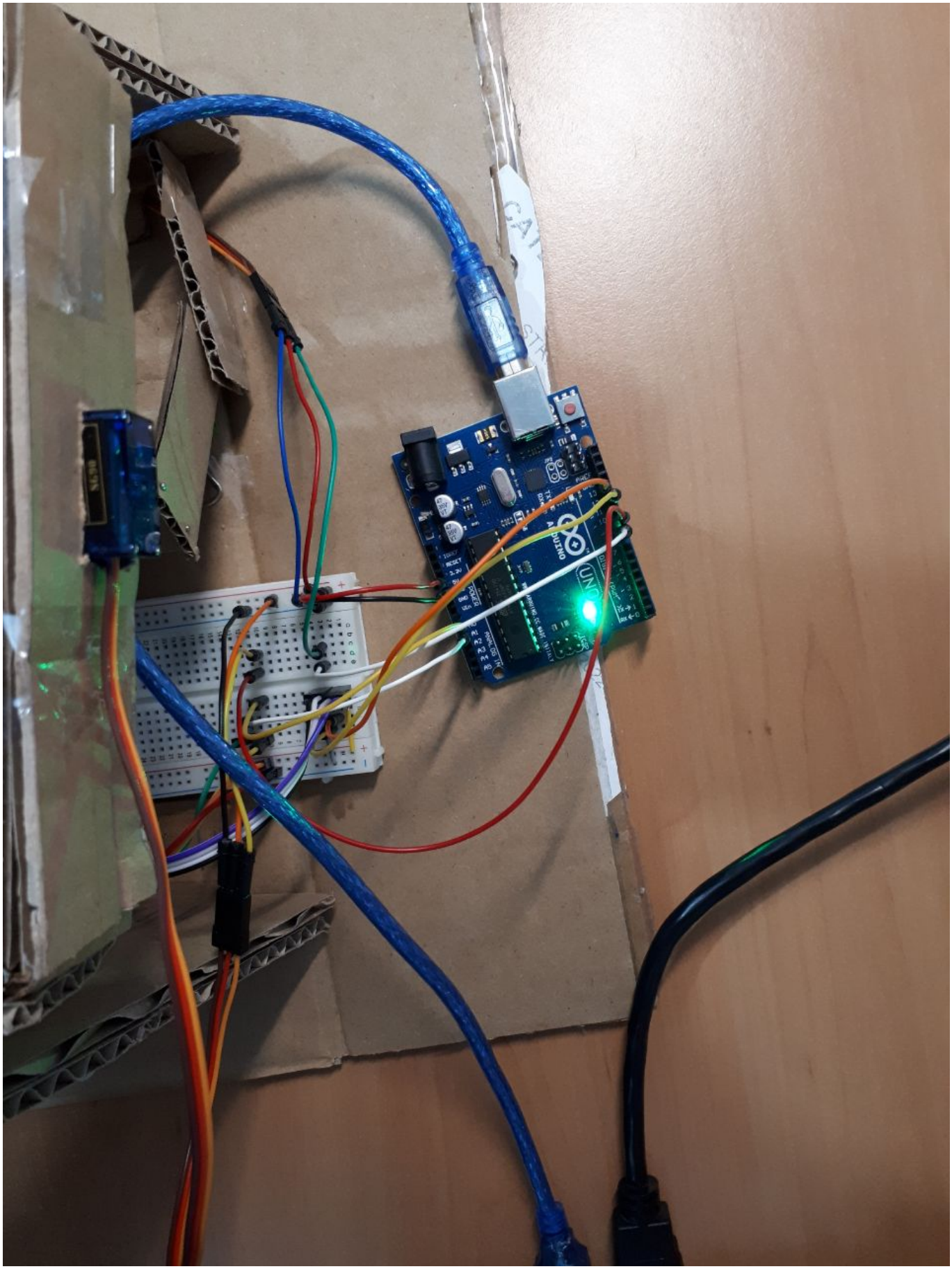
- n - Pornește un mod de joc simplu, fără contorizare de timp și fără funcționalitatea de detecție a bilei.
- 1 - Pornește un mod de joc contra timp, cu timpul limita de 120 de secunde si funcționalitatea de detecție a bilei activată.
- 2 - Pornește un mod de joc contra timp, cu timpul limita de 60 de secunde si funcționalitatea de detecție a bilei activată.
- 3 - Pornește un mod de joc contra timp, cu timpul limita de 30 de secunde si funcționalitatea de detecție a bilei activată.
- r - Aduce motoarele la poziția de start și oprește jocul curent.
- p - Afișează date utile pentru debug/setarea variabilelor în funcție de erorile fizice ale modelului construit.

Rezultate Obţinute

Am întampinat câteva probleme hardware pe parcurs, cum ar fi arderea unui servo motor sau erori de măsurare a distanței folosind senzorul ultrasonic, iar ca și probleme software, am fost nevoit să renunț la un buzzer pasiv deoarece timer2 interacționa cu funcțiile tone() și noTone().







Demo

Link demo YouTube

Concluzii

Având în vedere toate problemele întampinate și materialele folosite, proiectul a ieșit mai bine decât mă așteptam.

Am reușit să îmbin cunoștințe din laboratoarele de PM cum ar fi *Timer* pentru modurile de joc, interfața *Serială* pentru interacțiunea cu utilizatorul și *Debouncing* pentru lucrul cu servo motoare.

Download

Arhivă cod sursă
Export to PDF pagina Wiki

Jurnal

- 11.05.2022 Finalizare pagina Wiki
- 10.05.2022 Finalizare ansamblu hardware. Editare pagina wiki: Rezultate obtinute, Download
- 7.05.2022 Editare pagina wiki: Introducere, Software Design, Bibliografie
- 6.05.2022 Finalizare implementare cod, Update minor wiki + adaugare schema electrica EAGLE
- 21.04.2022 Completare pagina wiki: Introducere si Descriere generala
- 13.04.2022 Alegere tema proiect

Bibliografie/Resurse

Resurse Software

- https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
- https://www.tutorialspoint.com/arduino/arduino_ultrasonic_sensor.htm
- https://www.arduino.cc/en/Tutorial/BuiltInExamples/Debounce
- https://youtu.be/Uv9UeYUsA8A
- https://github.com/offtherails2010/Eagle-Custom-Libraries/blob/master/Ultrasonic%20Sensor%20HC-

SR04.lbr [pentru realizarea schemei Eagle]
- https://app.diagrams.net [pentru realizarea schemei bloc]

Resurse Hardware

- https://www.instructables.com/Arduino-Marble-Maze-Labyrinth/#discuss
- http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf
- https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf
- https://www.electroschematics.com/wp-content/uploads/2013/07/HC-SR04-datasheet-version-2.pdf